S-BPM ONE 2009

Albert Fleischmann 22. Oktober 2009

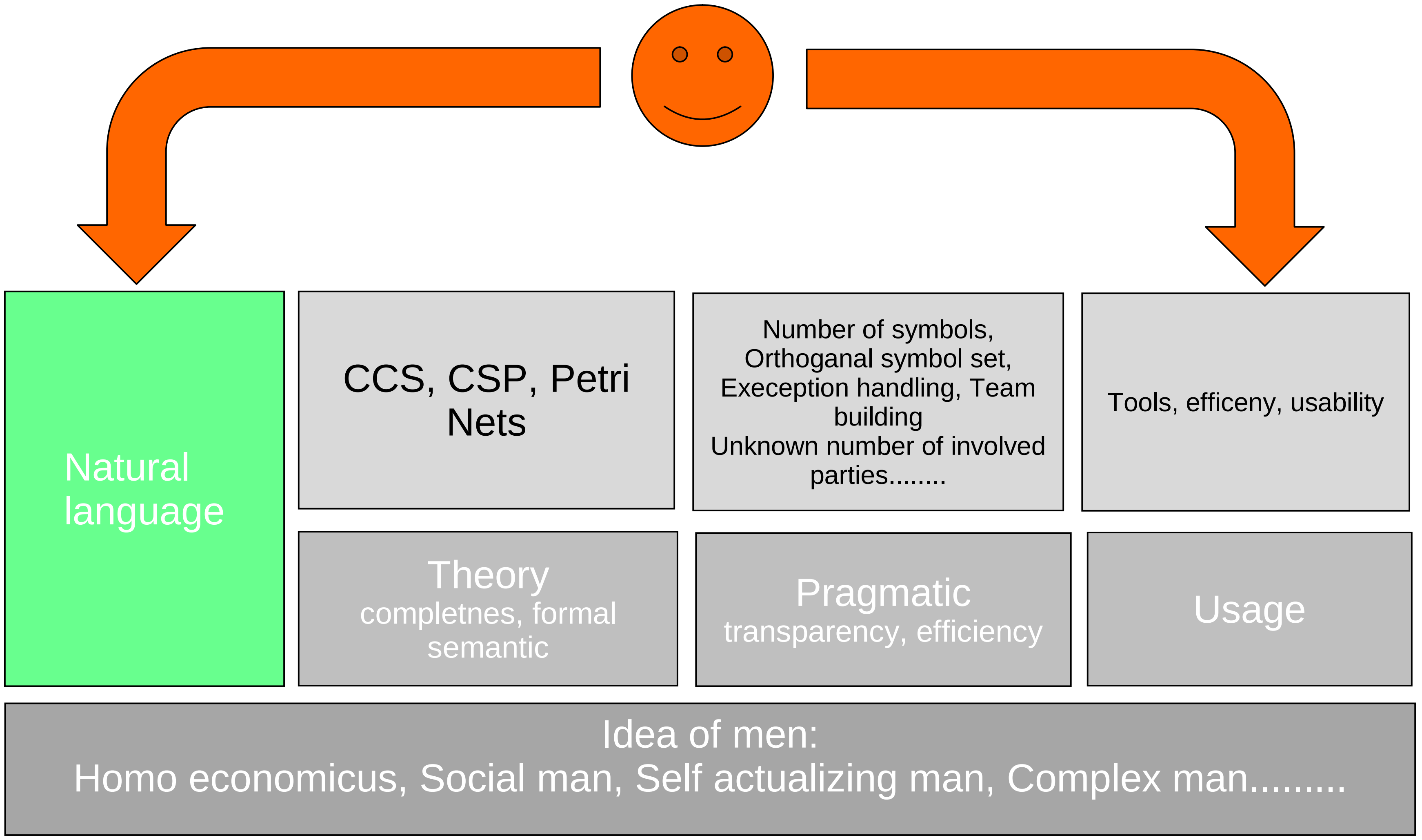
Agenda

- Properties of BPM 2.0
- Metamodel of BPM
- W-Questions meet Subject Predicate Object
- Various meanings of the word „subject“
- Focus of BPM modeling approaches
- Subjects define the granularity of actions in business processes
- CCS/CSP based S-BPM
- Ideas to subject oriented Petrinets
- Summary

BPM 2.0

- model can be built and adapted by process users.
- models can be executed without additional programming
- The process environment (People and Machines) can be easily integrated into the BPM model
- Process execution can be measured

BPM Metamodel



Natural language
CCS, CSP, Petri Nets
Number of symbols, Orthoganal symbol set, Exeception handling, Team building
Unknown number of involved parties........
Tools, efficeny, usability
Theory
completnes, formal semantic
Pragmatic
transparency, efficiency
Usage
Idea of men:
Homo economicus, Social man, Self actualizing man, Complex man.........

W-questions of Business Process Management

model aspects:

- Who are the parties involved in a Business Process? — Subject
- Which actions are executed in a Business Process? — Predicate — Sentence
- What are the target objects of actions? — Object
- When are the actions executed? — Story
- Why is a process neccessary?

Implementation aspects:

- Where are the parties involved in process located?
- How are the actions executed, Which machines and Application programms are used?

Different meanings of the word „Subject“

Differences in languages e.g. Englisch and German

Subjects in Mails

Subjects in Programming

Subjects in Logic

Subjects in Semantic Webs

Subjects in Philosophy

Subjects in Grammar

....

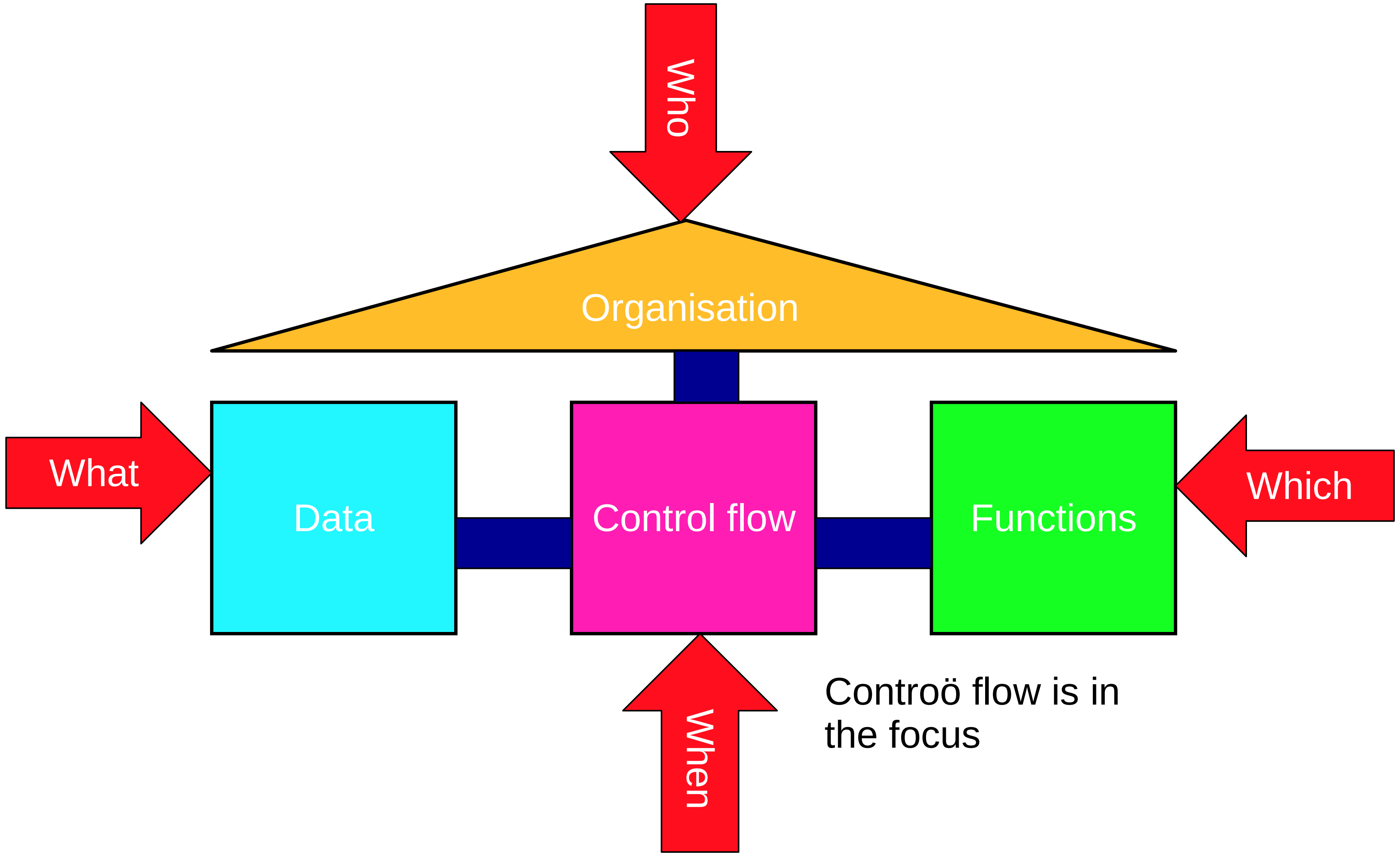
Subjects in subject oriented BPM or programming

- Definition close to the usage in grammar
- Subject are active elements in business processes
- Subjects are abstract resources which execute actions defined on objects
- Subjects synchronize their activities

Various modeling methods emphasis different aspects

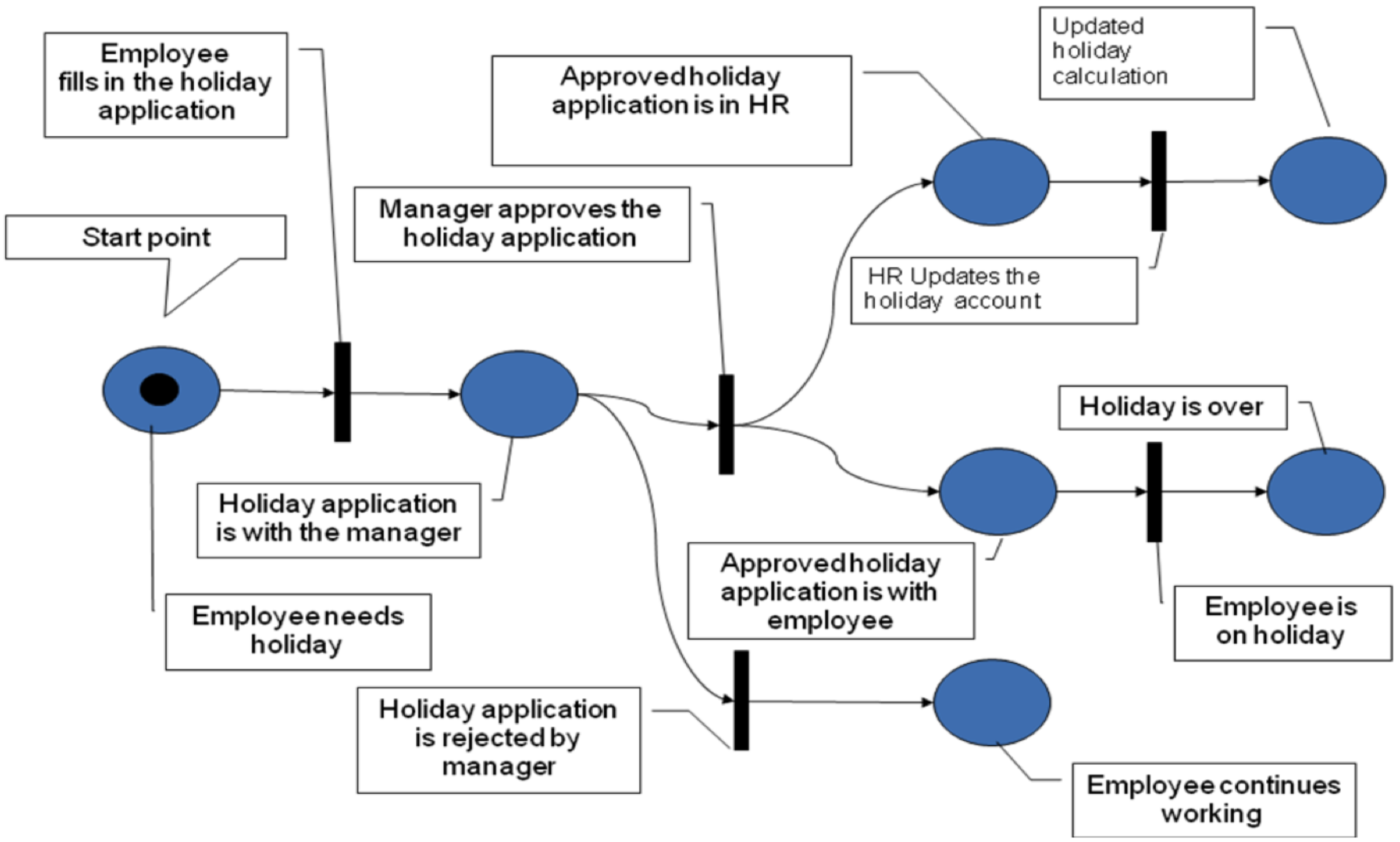
- Subject
    - CCS and CSP
    - Swim Lanes
    - PASS
- Predicate
    - Petrinets
    - EPK
    - UML
- Objects
    - ARS: Action Request System
    - Workflows in Document Management Systems (DMS)
    - UML

Who, Which, What, When in ARIS



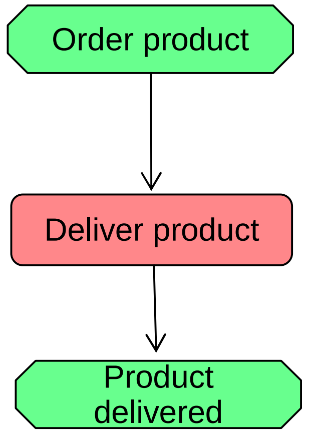
Controö flow is in the focus

Petri Nets

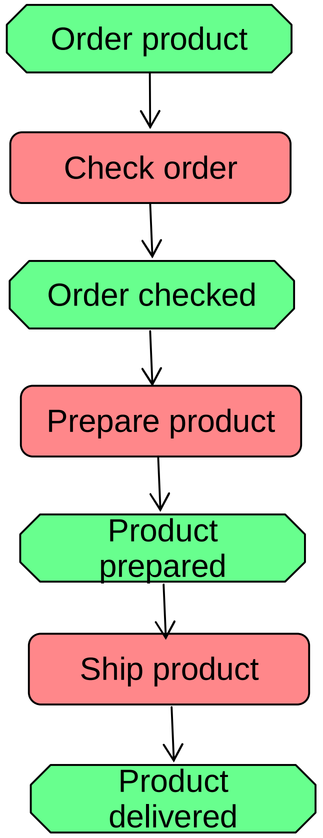

Updated holiday calculation
Employee fills in the holiday application
Approved holiday application is in HR
Manager approves the holiday application
Start point
HR Updates the holiday account
Holiday is over
Holiday application is with the manager
Approved holiday application is with employee
Employee is on holiday
Employee needs holiday
Holiday application is rejected by manager
Employee continues working

What is the starting point for process modeling

- Action
  - Problem: The required granularity of the actions can not be defined!
- Example

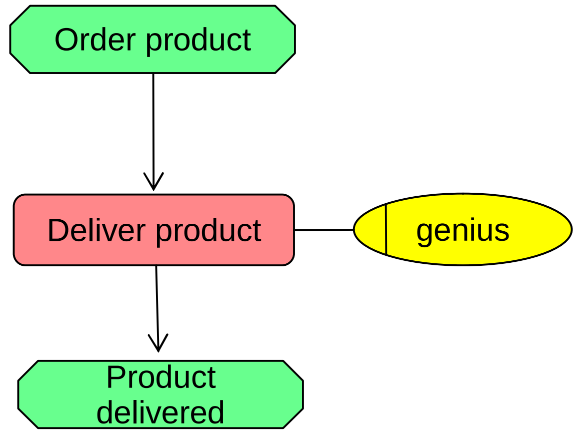


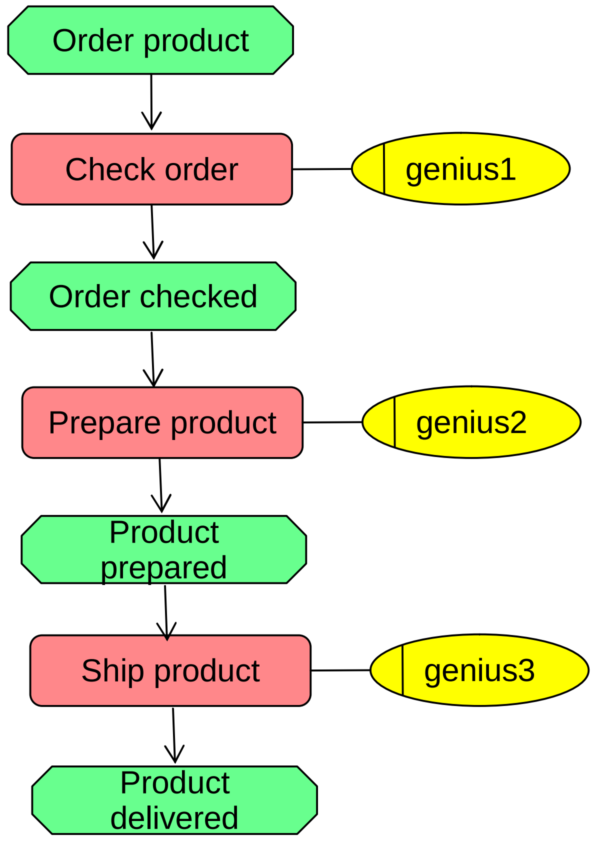
Which is the right granularity
?



S-BPM The leading part in a process are the subjects

- The subjects of a process define the granularity of the actions in a process!

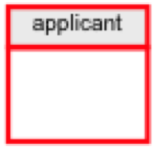
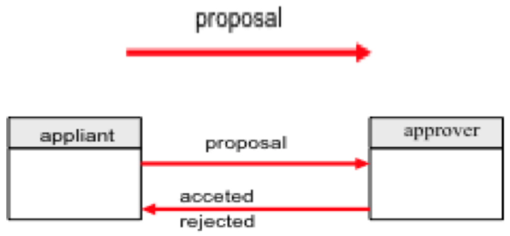
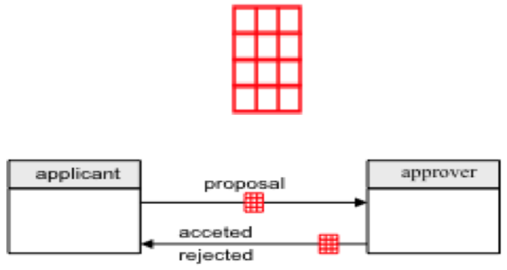




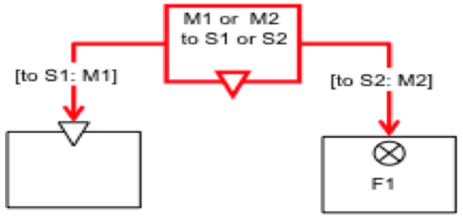
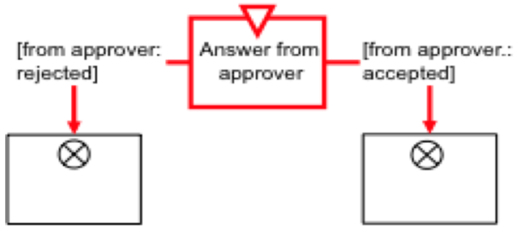
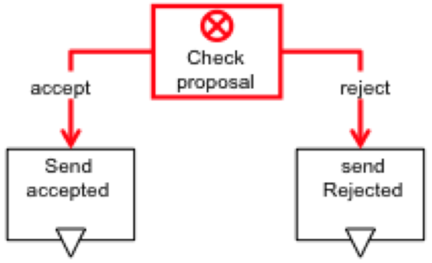
S-BPM based on CCS/CSP

- Identify your processes and create your process network
- Identify the subjects in a process
- Identify the messages exchanged between subjects
- Identify the payload of the messages
- Define the behaviour of each subject
- Embed your model in your environment

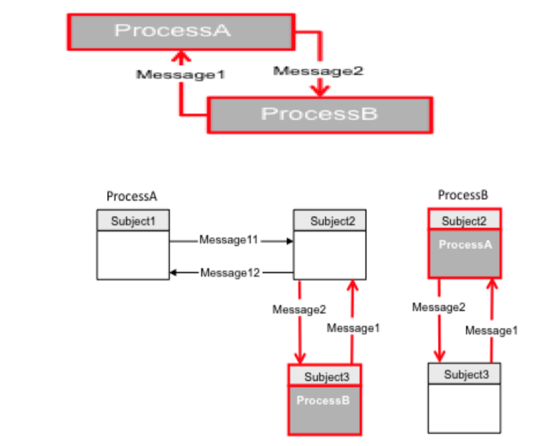
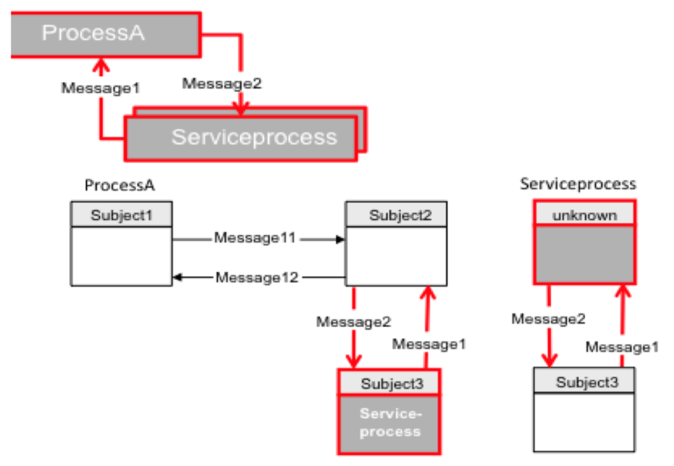
Basic modeling Elements: Communication Structure

	Subject	The acting elements in a process are called subjects. Subjects have a name, which describes its role in the process. Subjects send or receive messages and execute actions. The sequence in which they do these activities is described in their behaviour (see below).
 proposal appliant proposal approver acceted rejected	Message	Subjects exchange information and synchronize their activities by exchanging messages. Each message has a name. The message name should give an indication of the content and purpose of a message. Messages can be exchange asynchronously or synchronously. Each subject has an input pool in which the sending subject deposits the messages for that subject. The corresponding subject accepts messages by removing them from the input pool.
	Business Objects	Messages transport information as business objects. Each business object has a name and content. The name of a business object should give an indication about the purpose of a business object. Business objects are exchanged between subjects via messages. Business objects are the payloads of messages.

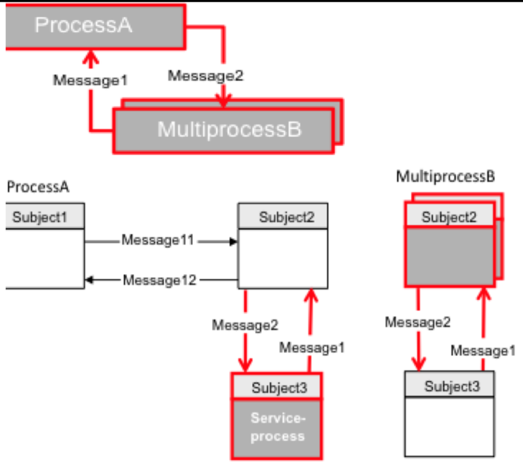
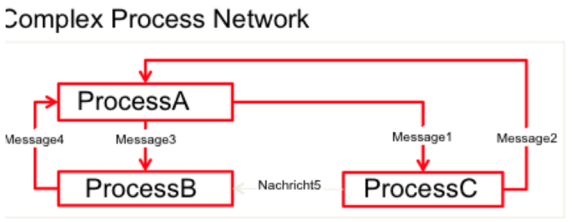
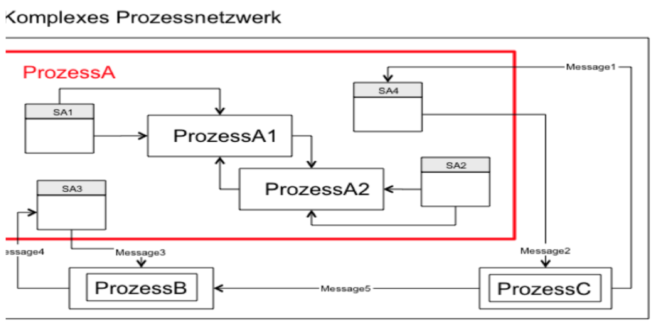
Basic modeling Elements: Behaviour

	Send messages	Subjects send messages to other subjects. In a send state, a subject tries to send several messages alternatively. In our example, a subject tries to send either message M1 subject S1 or message M2 to subject S2. If a message can be sent to the addressee the corresponding transition is executed.
 [from approver: rejected] Answer from approver [from approver.: accepted]	Receive Messages	Subjects receive messages from various subjects. In a receive state a subject can accept several messages alternatively. In our example, a subject waits for the message rejected from the subject approver and for the message accepted from the subject approver. If one of these messages is available, the corresponding transition is executed. If both messages are available than one will be selected randomly. “Message available” means that it has been sent by the corresponding subject.
 Check proposal accept reject Send accepted send Rejected	Actions	Subjects execute also internal actions. Internal actions are defined on internal data like business objects. The operation is executed in the state. The execution of an action can end up with several results. In our example, the action “Check Proposal” can have the result “accept” or “reject”.

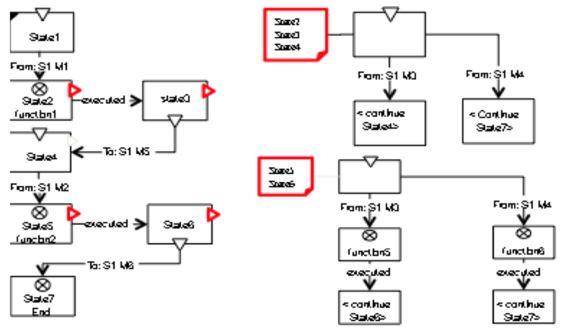
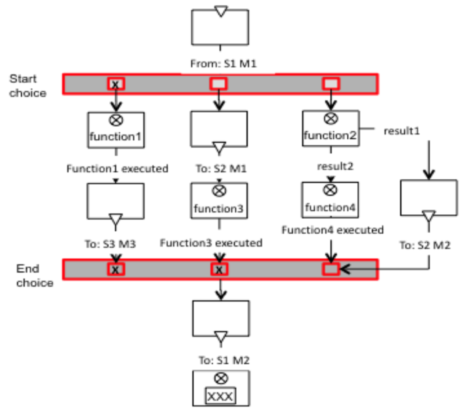
Pragmatic: Complex Process Structures 1

 ProcessA Message1 Message2 ProcessB ProcessA Subject1 Message11 Message12 Subject2 ProcessB Subject2 ProcessA Message2 Message1 Subject3 ProcessB Message2 Message1 Subject3	Connected Processes	If processes become to complex e.g. to many subjects, then processes can be separated in several connected processes. Subjects in the various processes connect these processes. Subjects in a process may interact with subjects in other processes. In the considered process, subjects interact with subjects in other processes. These external subjects are represented in the considered process. The example shows two connected processes: ProcessA and ProcessB. Subject2 in ProcessA interacts with Subject3 in ProcessB. Subject1 in ProcessA and Subject3 in ProcessB exchange the messages “message1” and “message2”. Subject3 is rpresented as an external subject in ProcessA and subject2 is represented as external subject in ProcessB (cross reference).
 ProcessA Message1 Message2 Serviceprocess ProcessA Subject1 Message11 Message12 Subject2 Serviceprocess unknown Message2 Message1 Subject3 Service-process Message2 Message1 Subject3	Service Processes	Service processes are very similar to connected processes. In a service process there is a subject which is visible for all other processes. This subject is called interface subject. This interface subject accepts messages from subjects in any other process. These subjects are unknown to the interface subject of a service process. Messages to an interface subject are service invocations. An interface subject accepts messages and stores the name of the sending subject in a variable. The variable is used to send messages to the invoking subject. In Service processes there is no cross-reference between these connected processes. A service process does not know which subjects in which processes send messages to it.

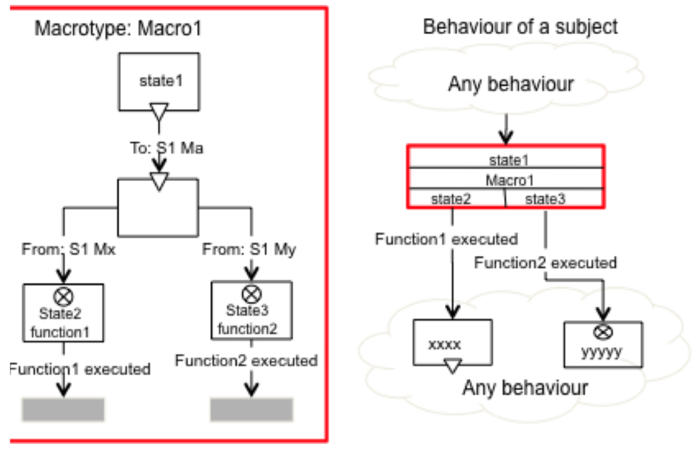
Pragmatic: Complex Process Structures 2

 ProcessA Message1 Message2 MultiprocessB ProcessA Subject1 Message11 Message12 Subject2 MultiprocessB Subject2 Message2 Message1 Subject3 Service-process Message2 Message1 Subject3	Multi-processes	Multiprocesses are an extension of connected processes. If a multiprocess receives a certain message from a connected process, a copy of the multiprocess is produced. Every time that message is sent to the multiprocess, a new copy is generated. In our example every time subject2 sends the message "message2" to the subject3 which is part of multiprocess "ProcessB" a new copy of ProcessB is produced. Multiprocesses allow to modell certain business cases very elegant. Examples of such cases are reviews of papers or request for proposals with a unknown number of reviewers or suppliers.
 Complex Process Network ProcessA Message4 Message3 Message1 Message2 ProcessB Nachricht5 ProcessC  Komplexes Prozessnetzwerk ProzessA Message1 SA4 SA1 ProzessA1 SA2 SA3 ProzessA2 Message4 Message3 Message2 ProzessB Message5 ProzessC	Hierarchical Process Networks	A process can be connected with an other process. This connected process can be connected with other processes and so on. In that way, hierarchical networks of connected processes can be constructed. Our example shows a network with three connected level one processes (ProcessA, ProcessB and ProcessC). Each process can consist of subjects and other processes. ProcessA consists of the processes ProcessA1 and ProcessA2. ProcessA1 and ProcessA2 are also connected. In our example, we show only the subjects in ProcessA, which are visible by subjects in other processes (external subjects). There can be any number of internal subjects, which are not seen by other processes.

Pragmatic: Complex Process Behaviour 1

	Exception Handling	Sometimes it is necessary to accept a certain set of messages in many states e.g. an order can be cancelled in any delivery state. Instead of adding the corresponding transitions to any concerned state, the handling of those messages is separated from the main activity sequence. In our example, two different handlers handle the messages M3 and M4 from subject S1. In one handler the states state2, state3, state4 are covered and the second handler covers the states state5 and state6. If one of the observed messages is available in one of the corresponding controlled states, the subject continues with the corresponding handler. At the end of each handler, it is defined in which state of the main behavior the subject continuous.
 From: S1 M1 Start choice function1 function2 result1 Function1 executed To: S2 M1 result2 function3 function4 Function4 executed To: S3 M3 Function3 executed To: S2 M2 End choice To: S1 M2 XXX	Choice	Sometimes it should be possible that actions are overlapped. This means thatthey can be excuted in any order. During execution, it can be choosen which actions are executed when. Our example shows such a choice operation. The open choice operator defines the number of overlapping sequences. In our case, there are three overlapping sequences. Each of these sequences has an active state. A subject can choose randomly which of the active states will be executed next. Normally a subject has only one active state. Some sequences in a choice operator must be executed (in our example the left path, marked with an x at the start and end of the sequence), others can be executed but if execution has started they must be finished (in our example the sequence in the middle, marked with an x at the end of the sequence) and others can be executed but must not be finished (right sequence, no marks). The subject can switch randomly between the active states of the sequences in a choice operator.

Pragmatic: Complex Process Behaviour 2

 Macrotype: Macro1 state1 To: S1 Ma From: S1 Mx From: S1 My State2 function1 State3 function2 Function1 executed Function2 executed Behaviour of a subject Any behaviour state1 Macro1 state2 state3 Function1 executed Function2 executed xxxx yyyyy Any behaviour	Macros	It happens that the same activity sequences are used in behavior specifications of several subjects. These sequences can be defined once as a so-called macro. Macros can be embedded in subject behaviors where the corresponding macro behavior is required. In our example we have defined a macro in which a message is send to a subject and to answers are expected. After each answer, a corresponding follow up activity is executed. This macro can be embedded in the behavior of any subject provided the partner subject is the same (in our case it must be subject S1). In order to define macros independent from communication partners it is possible to define macro types.

Subject based Petrinets

- Definition of Petrinets:
- Add Subject Mapping
- Add Object Mapping
- Define Subject Views
  - Consider only transitions assigned to certain subjects together with their input and output places: Deadlocks. Livelocks ....
- Investigate properties of subject oriented petrinets:
  - Livelocks
  - Deadlocks
  - ....
- Define process instances by coloured tokens
- Embedding of subject oriented Petrinets in their environment

Conclusion and Summary

- BPM 2.0 means agility, simplicity and direct execution of models
- Start with the basic building elements of all natural languages
- Map these elements to existing formal models or extend these formal models to have subject, predicate and object.
- The granularity of Actions in Business Process is defined by the involved parties
- Enrich these formal models with modeling elements für improving tranparency and efficiency
- Implement coreponding tools which are the entrance for the user.

Thank you for your attention

Questions?